Final Report

Senior Design - Spring 2018

Asset Management: Financial Factor Discovery - “Value”
Advisor: Chinmay Hegde
Client: Principal Global Investors

Team:

Caleb Utesch
Carter Scheve
Alex Mortimer
Nathan Hanson
Sam Howard
Jack Murphy

Principal℠

Project Plan

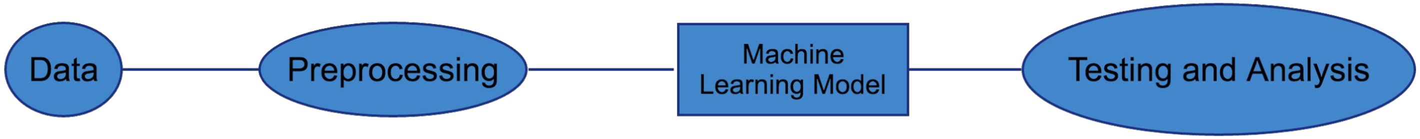
Problem Statement

- Current investment techniques rely on human analysis
  - Inefficient and error-prone
  - Client wants an automated system for predictions and suggestions
- Solution:
  - Software tool for making investment predictions
  - Utilize Machine Learning and Statistical Modeling
- Advantages
  - Eliminate erroneous human decisions
  - Increase decision-making speed for volatile market

Task Responsibility

- Carter Scheve — Communications Lead
  - Built, maintained and updated data library; researched Naive Bayes model
- Nathan Hanson — Project Progress Tracker/Manager
  - Support Vector Machine Classification and classification confidence
- Caleb Utesch — Meeting Scribe
  - Linear regression analysis and feature selection techniques
- Jack Murphy — Research Analyst
  - K-Nearest Neighbors and recursive feature elimination
- Samuel Howard — Lead Engineer
  - Autoregressive models and principle component analysis
- Alex Mortimer — Project Manager
  - Random forest classification and regression

Conceptual Sketch



- Data requires preprocessing to fit into model training, testing formats
    - Handling NaN values, separating factors, identifying features
- Models train on a subset of data, test on the remainder
- Accuracy results are analysed to find best model, feature list, parameters, etc.

Principal℠

Functional Requirements

- Models report processing time and total accuracy
- Results are displayed in a human-readable format
- Models only use past data to predict future trends
- Summary of models gives concrete statistics for performance of each individual model, along with a comparison of each and a recommendation for which to use in similar future tasks

Principal℠

Non-functional Requirements

- Use Python for development
- Utilize machine learning techniques to make predictions
- Model must be maintainable for the foreseeable future
- Easily integrated into current portfolio management systems at PGI

Technical/Other Constraints/Considerations

- Standards used:
  - ISO/IEC/IEEE 15939:2017(E) Reporting Standard
  - PEP8 Python conventions standard
- Considerations
  - Data that we are provided is sensitive company data, so it needs to be kept private
  - Many ethics considerations and other standards have already been taken care of by our client

Principal℠

Market Survey

- Predicting the direction of stock market prices using random forest
    - Paper published in 2016
    - In-depth discussion of the mechanics of Random Forest
    - Displayed very high accuracy for short term classification results
- Prediction Algorithms and Confidence Measures Based on Algorithmic Randomness Theory
    - Paper published in 2002
    - Introduction to confidence measures in classification models
    - Achieved about 99% accuracy in classifying handwritten digits using SVM with confidence measures.
- Principal Component Analysis
    - Paper created in 2017
    - Demonstrates the usefulness and potential for PCA
    - Shows Cross Validation techniques to improve models

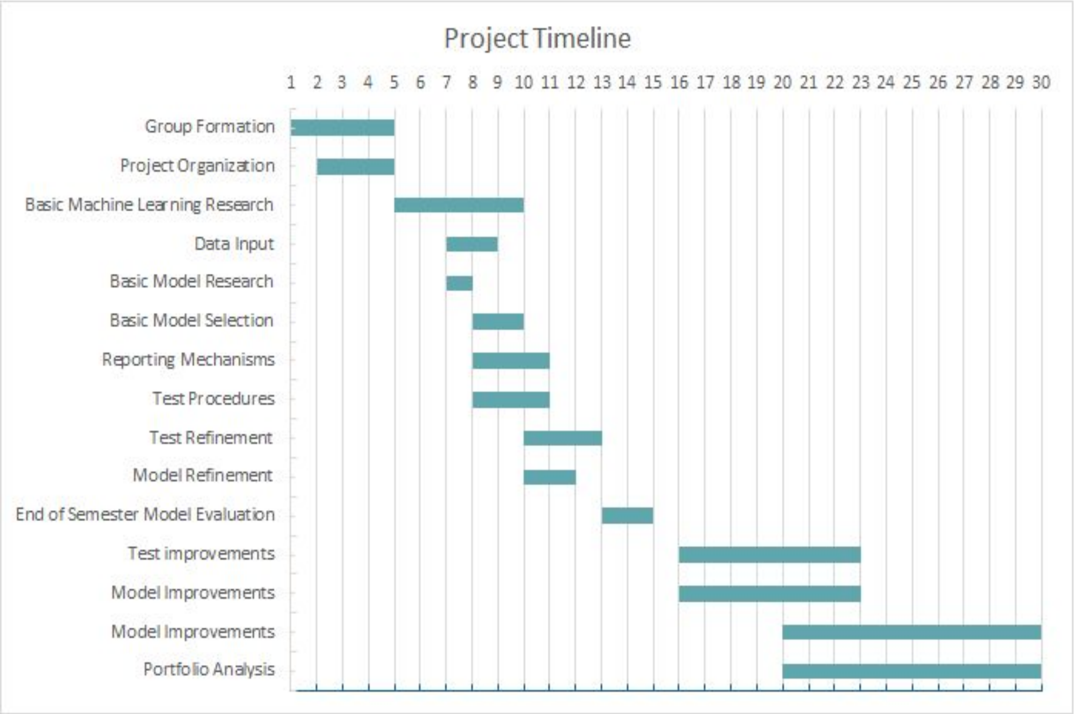
Potential Risks & Mitigation

- Several main risks include:
  - Lack of knowledge in area of machine learning
  - Models that are developed won't provide sufficient information for PGI

- Mitigation techniques:
  - Extensive research into machine learning models
  - Weekly meetings with client to try and ensure we are producing adequate results

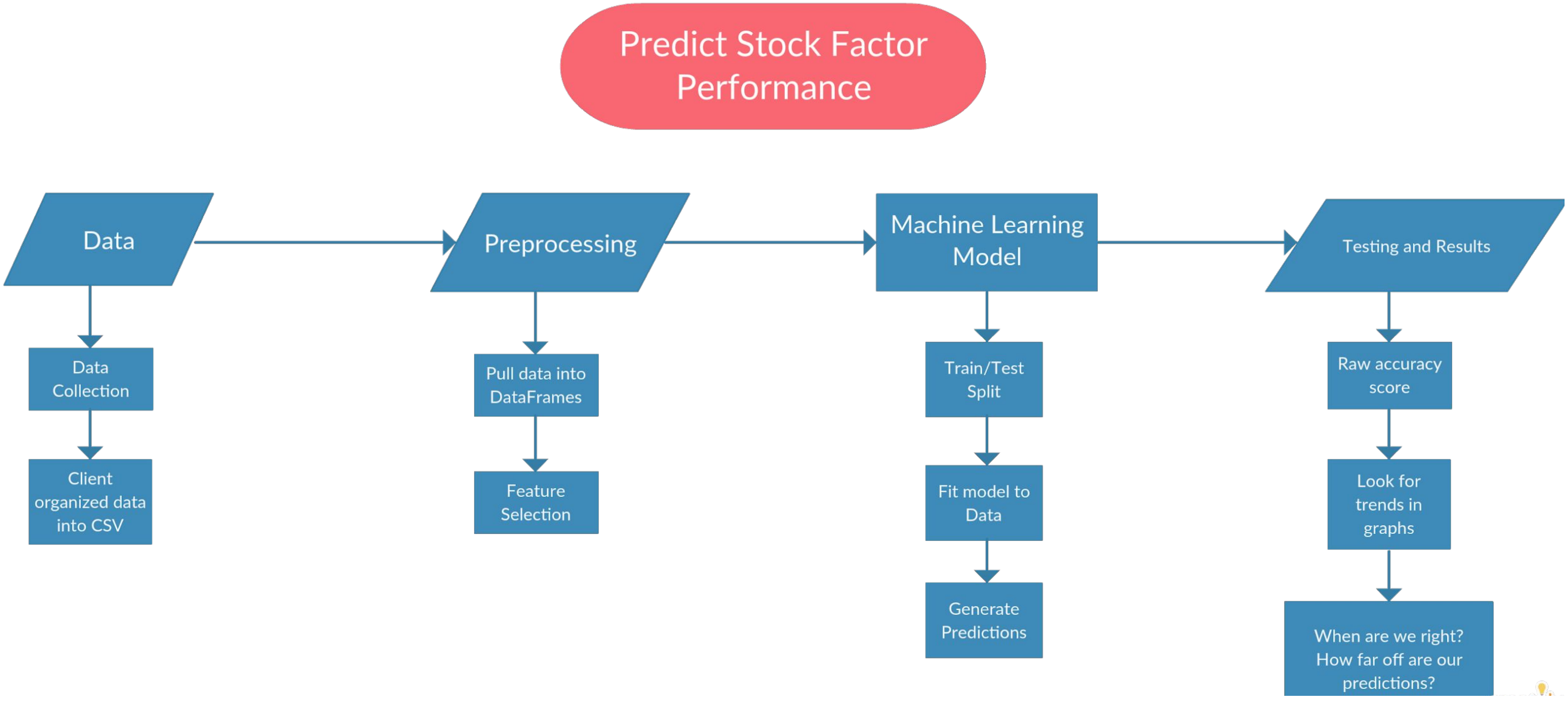
Project Milestones & Schedule

- Milestones:
  - Machine Learning Research
  - Model Development
  - Realistic Accuracy



System Design

Functional Decomposition



Detailed Design

1. Exploratory Data Analysis
    - Kaggle, Datacamp
2. Machine Learning Model Selection
    - Original models chosen: SVM, Naive Bayes, Random Forest, K-Nearest Neighbors, Autoregression
    - Current models in use: Naive Bayes, Random Forest
3. Feature Selection methods
    - Tree-based selection, recursive feature elimination, univariate selection, principal component analysis
4. Using sliding/Expanding Window test-train split, Regressive models

HW/SW Platforms used

- Python 3.6
- Numpy
- Pandas
- Matplotlib
- Scikit-learn
    - Machine Learning model implementations
    - Accuracy Reports

- No external hardware platforms used

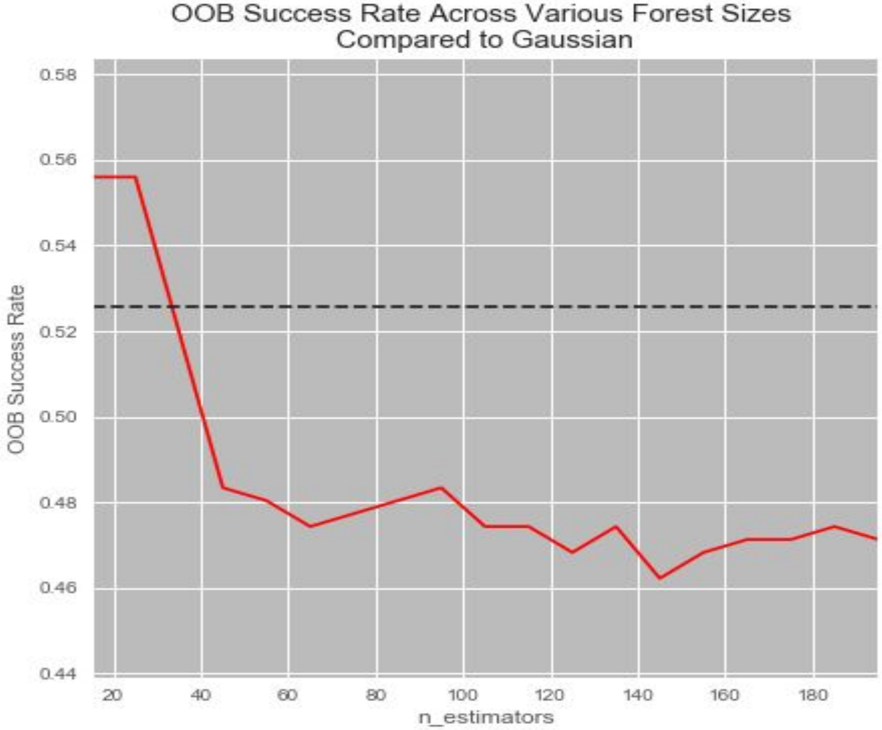
Test Plan

- Initial testing
    - Of popular and applicable machine learning classification and regression models, select those which best model the given market data.
    - Adjust model parameters to continue improving model fit and prediction accuracy.
- Portfolio Analysis
    - Using the chosen models and parameters, simulate an optimal portfolio basing decisions on model predictions.
    - Aim for models to consistently predict which stocks will outperform market.

Principal℠

Prototype Implementation

Random Forest Results
All Features



Principal℠

Prototype Implementation

Naive Bayes Results
All Features
Training set: 70%
Score: 0.519637462236

Classification Report

	Precision	Recall	f1-score	Support
underperform	0.73	0.31	0.44	198
outperform	0.45	0.83	0.58	133
avg	0.62	0.52	0.5	331

Principal℠

Conclusion

Principal℠

Current Project Status

- Basic models implemented
- Feature selection techniques started
- Framework for final model results and comparisons in place
- Data Importing library under revision to handle new data set

Principal℠

Plan for next semester

- New dataset
- Restart
    - Models
    - Experiments and Exploration
    - Feature Engineering
- Advantages
    - More data
    - Experience
    - Speed

Thank you!

Random Forest (classification)

- Predictive model that creates decision trees based on training data to predict whether a factor outperforms or underperforms the market
- Best set of features
    - 'SALES_P_Volatility',
    - 'EBIT_MCAP_Bat',
    - 'X9M_RET_Volatility',
    - 'CFO_P_Volatility'
    - 'ROE_Bat',
    - 'DIV_YID_Bat',
    - 'SALES_EV_Bat',

Random Forest (regression)

- Predictive model that creates decision trees based on training data to predict the actual future value of a certain feature in the market
- Average r-squared value: -2.58
- Raw model score: -0.79
- Predictor: RET_F12M
- Features Used are the same as above

Principal℠

Naive Bayes (classification)

- Predictor: RET_F12M_OP
- Current Features
    - From Univariate Selection
    - X6MVT, X12MVT, BETA_1Y, NET_CFO_P, BK_P, SALES_P, AST_P, OIL_SEN, X12MVT_Med, BAA.AAA_Mkt
- Overall Accuracy: 73.4375%
    - Train Size: 75%
- Expanding Window: 7 Years
    - Avg Accuracy: ~52%
    - Range: 38%-70%
- Sliding Window: x Years
    - 1 Year test window
    - Avg Accuracy: ~62%
    - Range: 25%-95%

Tree-based selection (feature selection)

- 11 features currently
  - FCF_P_Bat
  - SALES_P_Volatility
  - EBIT_MCAP_Bat
  - X9M_RET_Volatility
  - CFO_P_Volatility
  - ROE_Bat
  - DIV_YID_Bat
  - SALES_EV_Bat
  - TED_SEN_Volatility
  - X6M_RET_Bat
  - RANK_Bat

Principal℠

Tree-based selection (feature selection)

- 11 features currently
  - FCF_P_Bat
  - SALES_P_Volatility
  - EBIT_MCAP_Bat
  - X9M_RET_Volatility
  - CFO_P_Volatility
  - ROE_Bat
  - DIV_YID_Bat
  - SALES_EV_Bat
  - TED_SEN_Volatility
  - X6M_RET_Bat
  - RANK_Bat

Recursive feature elimination (feature selection)

- 10 features of highest rank
  - D_E
  - SALES_AST
  - Val_SD_Mkt
  - X12MVT_Bat
  - X1SS_ERNQLT
  - X3IVH_CGBS
  - X3IVH_CGBS_Bat
  - X6MVT_Bat
  - X6M_RET
  - X9M_RET

Principal℠

Autoregression (predictive model)

- Uses pattern of past output points to determine the future. Useful when future events can be preceded by the relatively recent past.
- The data is autocorrelated, with partial autocorrelation tests showing an optimal A value of 2.
  - Model completely nonviable, given we are predicting 52 weeks out, and the model needs to generate and use every intervening week for the next one

Principal℠

Univariate selection (feature selection)

- 10 features currently
  - X6MVT
  - X12MVT
  - BETA_1Y
  - NET_CFO_P
  - BK_P
  - SALES_P
  - AST_P
  - OIL_SEN
  - X12MVT_Med
  - BAA.AAA_Mkt

L1-based selection (feature selection)

- 18 features currently
  - X1,3,6,9,12M_RET
  - MCAP
  - D_E
  - ROE
  - SALES_AST
  - FY1_3MCHG
  - X3IVH_CGBS
  - X1SS_ERNQLT
  - BAA.AA_Mkt
  - ti.rank_Mkt
  - cor.rank_Mkt
  - Val_SD_Mkt
  - Crowd_Mkt
  - Earnings_Res_Mkt

Principal component analysis (feature selection)

- Creates a new set of orthogonal axis to explain the maximum amount of variance in the dataset. Can be visualized as a rotation and translation of the axis.
- Over 99% variance using 15 components on normalized data
- Directions of maximum variance (decreasing order)
    - SALES_AST, MCAP, Earnings_Res_Mkt, D_E, cor.rank_Mkt
    - Crowd_Mkt, Val_SD_Mkt, X1SS_ERNQLT, X3IVH_CGBS
    - ti.rank_Mkt

Principal℠

Expanding Window Testing

- Expands the test window for the data set every iteration
- Expands test window by the test size
- Test size is set, training size increases

Sliding Window Testing

- Slides the full window for the data set every iteration
- Slides full window by the test size
- Test and train size is set, but the total window slides

Weighting

- Newer data has more effect than older data
- Weights for each data point can affect the accuracy of a model
- Different weight curves can affect the accuracy